REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
APRIL 2012	CONFERENCE PAPER (Post Print)	JAN 2009 – DEC 2010

4. TITLE AND SUBTITLE

BLIND EXTRACTION AND SECURITY ANALYSIS OF SPREAD SPECTRUM HIDDEN WATERMARKS

5a. CONTRACT NUMBER
FA8750-07-C-0199

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

John A. Marsh (AIS) and Gerard F. Wohlrab (AFRL)

5d. PROJECT NUMBER
231G

5e. TASK NUMBER
MM

5f. WORK UNIT NUMBER
JM

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Assured Information Security, Inc.
153 Brooks Rd.
Rome, NY 13441

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/Information Directorate
Rome Research Site/RITE
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2012-020

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA Case Number: 88ABW-2011-2497
DATE CLEARED: 24 APRIL 2012

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In the area of covert network communications, the focus has been on spread spectrum (SS) techniques using correlated host data, applicable to many data hiding and covert communications applications. Our work relates to the Iterative Generalized Least Squares (IGLS) blind signature recovery algorithm of Gkizeli et al.1, and can be summarized as follows: (1) We have performed extensive Monte Carlo simulations that characterize the convergence properties of the algorithm as a function of signature length, host distortion, and number of hidden bits; (2) We have developed and characterized the behavior of a fully blind extension of the IGLS algorithm, called the BC-IGLS (Blind IGLS using Clustering); (3) We have developed and performed a characterization study of an extension to the IGLS algorithm, called the MS-IGLS (Multi Signature IGLS), that performs blind extraction of multiple signatures in multi-user embedding applications.

15. SUBJECT TERMS
Spread Spectrum Hiding, Multi Signature Embedding, Watermarking, DSSS.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			MICHAEL J. MEDLEY
U	U	U	UU	15	19b. TELEPHONE NUMBER (*Include area code*) N/A

Blind extraction and security analysis of spread spectrum hidden watermarks

John A. Marsh[*a,b], and Gerard F. Wohlrab[c]
[a]SUNY Institute of Technology, 100 Seymour Dr., Utica, NY 13502;
[b]Assured Information Security, Inc., 153 Brooks, Rd., Rome, NY 13441;
[c]Air Force Research Laboratory, 525 Brooks Rd., Rome, NY 13441

ABSTRACT

In the area of covert network communications, the focus has been on spread spectrum (SS) techniques using correlated host data, applicable to many data hiding and covert communications applications. Our work relates to the Iterative Generalized Least Squares (IGLS) blind signature recovery algorithm of Gkizeli et al.[1], and can be summarized as follows: (1) We have performed extensive Monte Carlo simulations that characterize the convergence properties of the algorithm as a function of signature length, host distortion, and number of hidden bits; (2) We have developed and characterized the behavior of a fully blind extension of the IGLS algorithm, called the BC-IGLS (Blind IGLS using Clustering); (3) We have developed and performed a characterization study of an extension to the IGLS algorithm, called the MS-IGLS (Multi Signature IGLS), that performs blind extraction of multiple signatures in multi-user embedding applications[2].

Keywords: Spread Spectrum Hiding, Multi Signature Embedding, Watermarking, DSSS

1. INTRODUCTION

We consider here a particular form of covert communication: direct sequence spread spectrum (DSSS) data hiding[1] in digitized analog data, referred to as the host data. This form of covert messaging is often considered to be part of the field of watermarking, especially when applied to data hiding in compressed images. The technique is, however, more broadly applicable, and can be used in a variety of ways to hide data in multimedia files or multimedia data streams. The only requirement is that some level of distortion of the original signal is acceptable by the end-user. In the case of data hiding into compressed images, the DCT (Discrete Cosine Transform) coefficients used in JPEG compression offer such a dataset.

Spread spectrum data hiding proceeds by vector addition of a scaled copy of a signature vector to a block of host data of the same length. The signature vector is scaled by two components: the sign of the bit to be hidden (as a duobinary value, +1 or -1), and the embedding amplitude.

The most straightforward method of extracting hidden data in spread spectrum hiding is called the Matched Filter (MF) technique[2], commonly used in array processing and Code Division Multiple Access (CDMA) schemes for wireless communications. The matched filter extracts one bit as the sign of the inner product between the signature and one block of host data.

However, spread spectrum techniques are complicated by the presence of correlations in the host data, in which the autocorrelation matrix differs significantly from the identity matrix. The optimal extraction process is then the Minimum Mean Squared Error (MMSE) technique[3], which makes use of the autocorrelation matrix of the host data. As the correlations become stronger (and the autocorrelation matrix deviates more from the identity matrix), the MMSE technique increasingly outperforms the MF extraction technique. Digitized analog data often displays non-zero

* Corresponding author. john.marsh@ieee.org. This work was supported in part by AFOSR under Grant FA8750-07-C-0199.
Approved for public release, distribution unlimited: 88ABW-2012-2497 24 April 2012.

correlations, including the important case of DCT coefficients in JPEG images. Thus, our work considers the general problem of data hiding and extraction with correlated host data, for which some level of distortion is acceptable.

A problem of considerable practical importance is blind recovery, in which data extraction is attempted in the case of an unknown signature. Our focus is on characterizing the behavior of, and extending, a recent algorithm[4-7] that performs blind signature (and hidden data) recovery in fixed-signature spread spectrum (SS) hiding. The algorithm, known as the Iterative Generalized Least Squares (IGLS) method, is initialized with a guess at either the signature or the hidden data, then iterates, alternately using least squares estimates of the signature and the hidden data. Convergence of the algorithm to the proper signature and hidden data set is in general highly dependent on the initial guess. This series of studies also includes extensions to the IGLS algorithm using techniques similar to those presented here. The present study can be considered complimentary to these developments, due to differences in performance analysis and cover data sets used, as described below.

In addition to introducing the IGLS algorithm, the same group has more recently demonstrated optimal signatures[8-9], as eigenvectors of the cover data autocorrelation matrix. These signatures yield maximum capacity embedding, in the sense that, for a given host data set and recovery error rate, the embedding level is minimized. This same study considers the case of multiple signature embedding, and develops the optimal embedding scheme using multiple eigenvectors.

In what follows, we first set up the notation for the SS hiding problem, then report on the IGLS algorithm performance, for both real-valued and binary signatures. Using the known hidden message, we report qualitative results of measuring the fraction of runs that yield the correct hidden data, as a function of the number of hidden bits, the signature length, and the embedding amplitude.

Next, we look at the performance of our extension of IGLS, called BC-IGLS, which uses repeated IGLS runs and clustering to perform blind extraction of DSSS signatures and hidden data. We report detection theory metrics on the performance of this algorithm

Finally, we consider the use of IGLS recovery under multi signature embedding, using an extension to IGLS we call MS-IGLS. Here different embedding signatures are used simultaneously, and our algorithm sequentially extracts these multiple hidden data sets. The algorithm extension assumes the signatures used for multiple message SS hiding are orthogonal, and works by orthogonalizing the signature iterates to the entire set of previously found signatures. We present results showing the performance of this algorithm when used with random real-valued signatures, and also with optimal eigen signatures (described below).

2. THEORY

2.1 Direct Sequence Embedding and Extraction

We consider a set of N bits to hide, denoted by the 1xN column vector $\mathbf{b} \in \{\pm 1\}^N$. We perform hiding into a set of host data vectors $\mathbf{x}_i \in \Re^L, i = 1, \ldots, N$ using real-valued signature $\mathbf{s} \in \Re^L$ or binary signatures $\mathbf{s} \in \{\pm 1\}^L$. The hiding operation for the jth bit is performed using direct sequence spread spectrum (DSSS) embedding[1]

$$\mathbf{y}_j = \mathbf{x}_j + A b_j \mathbf{s} \tag{1}$$

where A is the embedding amplitude. If we define the matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \Re^{LxN}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \Re^{LxN}$ we can write the embedding for the entire data set in matrix form as

$$\mathbf{Y} = \mathbf{X} + A\mathbf{s}^T\mathbf{b} \tag{2}$$

Note this definition implies the j-k element of $\mathbf{X}$ is given by $\mathbf{X}_{jk} = (\mathbf{x}_k)_j$. We refer to $\mathbf{X}$ as the host data, or cover data.

We now consider the problem of extracting the hidden dataset from **Y**. We seek the optimal linear filter **w** that yields the best possible estimate $\mathbf{b}' = \mathbf{w}^T\mathbf{Y}$ of the original hidden data **b**. In the case where the host data vectors $\mathbf{x}_i$ are uncorrelated, the optimal filter for data extraction is the matched filter (MF), in which each bit is taken as

$$b_j' = \text{sign}(\mathbf{s}^T\mathbf{y}_j). \tag{3}$$

This can be written in matrix form as

$$\mathbf{b}' = sign(\mathbf{s}^T\mathbf{Y}). \tag{4}$$

This equation is interpreted as stating that the optimal linear filter, which acts on **Y** to extract the hidden data set **b**, is the signature **s** itself. When the host data is correlated the minimum mean squared error (MMSE) filtering provides optimal data extraction, given by

$$\mathbf{b}' = sign(\mathbf{s}^T\mathbf{R}_x^{-1}\mathbf{Y}) \tag{5}$$

where $\mathbf{R}_x = \frac{1}{L}\mathbf{X}\mathbf{X}^T$ is the L x L autocorrelation matrix of the host data set **X**. In other words, the linear filter acting on **Y** which extracts hidden data $\mathbf{b}'$ with the lowest bit error rate (BER) is $\mathbf{R}_x^{-1}\mathbf{s}$. Note MMSE filtering reduces to MF filtering in case correlations are not present in the host data.

2.2 Multiple Signature Embedding

Here we consider the situation of multiple-signature embedding into correlated host data. We embed K bitstreams $\mathbf{b}_k \in \{\pm 1\}^N, k = 1, \dots, K$. First we write the equation describing hiding of the j^{th} bit of all K bitstreams into the j^{th} host vector $\mathbf{x}_j$

$$\mathbf{y}_j = \mathbf{x}_j + \sum_{k=1}^{K} A_k b_{kj}\mathbf{s}_k. \tag{6}$$

Here A_k and $\mathbf{s}_k$ are the embedding amplitude and signature, respectively, of the k^{th} data set. Following the treatment above (for the case of single-signature hiding), we can write a matrix equation

$$\mathbf{Y} = \mathbf{X} + \sum_{k=1}^{K} A_k\mathbf{s}_k^T\mathbf{b}_k, \tag{7}$$

that describes the entire embedding process. In calculating the error probability, the idea is to consider one signal at a time, and treat the other signals as interference in much the same way as the noisy host data is treated.

2.3 Optimal Signatures and Maximum Capacity Embedding

We now consider the optimal signature, which yields, for a given host dataset **X**, the lowest possible error probability for a given embedding amplitude. This signature is known[1,2] to be the minimum eigenvalue eigenvector of the autocorrelation matrix of the host data. Indeed, let the eigenvectors of $\mathbf{R}_x$ be $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_L$ (with $\mathbf{q}_j^T\mathbf{q}_j = 1$ for all j) corresponding to the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_L$, where $\lambda_i \geq \lambda_j$ for $i < j$. Note that the eigenvectors of the matrix $\mathbf{R}_x^{-1}$ are $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_L^{-1}$, with the same corresponding eigenvectors. When using the eigenvector $\mathbf{q}_j$ as the embedding signature, a straightforward calculation yields

$$SNR = A^2\mathbf{q}_j^T\mathbf{R}_x^{-1}\mathbf{q}_j = \frac{A^2}{\lambda_j} \tag{8}$$

for each $j = 1, \dots, L$. This in turn yields the theoretical lowest possible BER for Gaussian cover data using the standard error function calculation[3]. We have used the fact that the eigenvectors are normalized such that $\|\mathbf{q}\| = \mathbf{q}_j^T\mathbf{q}_j = 1$. Using Eq. (3), and the fact that $\mathbf{q}_j$ is an eigenvector of $\mathbf{R}_x^{-1}$, we see that optimal MMSE filtering for eigen-signatures reduces to MF filtering.

For a given host dataset **X**, the lowest possible error probability for a given embedding amplitude is obtained using a signature that is the minimum eigenvalue eigenvector of the autocorrelation matrix of the host data. Eigenvector

signatures can also be used in a multiple-signature hiding scenario. One possibility is to use equal embedding amplitudes. However, to optimize the information hiding capacity at each step in the embedding, embedding amplitudes are varied according to eigenvalue number, as described in reference 2.

2.4 IGLS Algorithm for Blind Extraction

The IGLS algorithm[1,2] is based on the mean square error (MSE) technique for finding the optimal filter. For example, when we write $\mathbf{b}' = sign(\mathbf{s}^T\mathbf{R}_x^{-1}\mathbf{Y})$ as in Eq. (3) above, it means that $\mathbf{R}_x^{-1}\mathbf{s}$ is the optimal filter satisfying

$$\mathbf{w}_{MMSE} = \arg\min_{\mathbf{w}}(\|\mathbf{w}^T\mathbf{Y} - \mathbf{b}\|^2) = \mathbf{R}_x^{-1}\mathbf{s}. \tag{9}$$

The IGLS algorithm thus starts with a guess at the signature, and finds the optimal filter yielding the first guess at the data using that signature. The first step of the IGLS algorithm is to make an initial guess $\mathbf{s}_{\text{guess}}$ at the signature, and use it to extract a guess $\mathbf{b}'$at the hidden data

$$\mathbf{b}' = sign\left(\mathbf{s}_{guess}^T\mathbf{R}_y^{-1}\mathbf{Y}\right). \tag{10}$$

The next step of the IGLS algorithm performs a similar MSE calculation of the signature given the current guess at the data, $\mathbf{b}'$. We can write this as

$$\mathbf{s}' = \arg\min_{\mathbf{s}}\left(\left\|\mathbf{R}_x^{-\frac{1}{2}}\left(\mathbf{Y} - \mathbf{s}\mathbf{b}'^T\right)\right\|^2\right) = \frac{1}{N}\mathbf{Y}\mathbf{b}'. \tag{11}$$

This procedure is iterated alternately calculating the next guess at the data $\mathbf{b}'$, then the next guess at the signature $\mathbf{s}'$.

Note the IGLS algorithm works slightly differently when binary signatures are used, versus the more general real-valued signatures. If a binary initial signature guess is given, then the algorithm assumes binary signatures are being used, and Eq. (11) is replaced with

$$\mathbf{s}' = \text{sign}\left(\frac{1}{N}\mathbf{Y}\mathbf{b}'\right). \tag{12}$$

The IGLS algorithm is not guaranteed to converge to the correct signature/dataset. In fact, our study shows that the output of the IGLS algorithm depends strongly on the initial signature guess. In blind operation, the actual signature and actual hidden bits are unknown, so the core IGLS algorithm must be enhanced with some way of determining when it is likely that the correct signature and hidden data set have been found. As part of this work, an effective technique has been developed, which we now describe.

2.5 Blind IGLS Extraction Using Clustering (BC-IGLS)

When performing blind extractions we do not know if we have found the correct signature and corresponding data set. A natural solution is to perform multiple IGLS trials, each using a different initial signature guess, and look at the results in aggregate. Assuming the IGLS algorithm has some tendency to converge to the correct signature, with enough trials (depending on the parameters used in hiding) we can expect convergence to the correct signature/data two or more times, and the similarity between these two results will enable us to identify those cases where correct convergence has occurred.

Thus we consider the results of an IGLS trial to be a point in abstract space, and look for two or more results that are "close" to one another. In practice, we can consider either signatures or extracted data sets as the points in our abstract space. Our numerical experiments show that using the signature is the best option. This is mainly because it is much less computationally intensive when calculating pairwise distances for clustering (signatures are significantly smaller than the data sets, $L << N$).

An intuitive argument explains why this technique is viable. If the algorithm does show a tendency toward convergence, and if two initial signatures do converge to the correct solution, or something close, the distance between these signatures is expected to be small, an outlier in the set of distances between extracted signatures. In other words, if there is even a small propensity for the algorithm to converge, then the first two times it does will yield signatures very close (if not equal) to each other, much closer than the typical distance between two signatures when IGLS does not converge to the correct result, assuming these are uniformly distributed. This is the basis for our algorithm, and many others that use clustering as a means to find positive results.

Our code implements this scheme by using *k*-means clustering, where signature vectors are considered points in an abstract space. The *k*-means clustering proceeds by first calculating all pairwise distances between vectors. For binary vectors, Hamming distance is used, and for real-valued vectors the mean squared distance is used. It is possible to use other metrics as well. A minimum threshold distance is used to determine clustering: only if the distance between two vectors is less than the threshold will they be considered neighbors. A cluster of signatures is defined as the set of vectors that are all neighbors of some central vector. Our implementation by default returns just the largest cluster by simply finding the vector with the most neighbors. A variation returns all clusters with size above a minimum cluster size. If no two vectors are neighbors, then the algorithm returns no clusters.

The resulting algorithm, combining *k*-means clustering with the IGLS iterative technique, is referred to as blind IGLS extraction using clustering (BC-IGLS). The algorithm has as parameters the number of trial signatures, the minimum cluster size and the minimum threshold distance. Implementation details and test results for optimal parameters are described below.

2.6 Multi Signature IGLS (MS-IGLS) Blind Extraction

If multiple signatures have been used to hide multiple datasets, then a couple of obvious extensions to the BC-IGLS algorithm exist. First, multiple extractions can be made by repeatedly applying BC-IGLS and hoping that multiple initial signature guesses will yield multiple clusters, finally yielding all signatures. The emergence of clusters depends strongly on the embedding amplitude, with signatures that use larger embedding amplitude appearing preferentially. Thus it may be very difficult, using this technique, to ever arrive at signatures with weakest embedding amplitudes. A second alternative would be to repeatedly apply BC-IGLS extraction and subtract out the embedded data each time a new signature is found. This technique is not expected to work in a straightforward implementation, however, because the embedding amplitudes are a priori unknown.

We have developed an algorithm that is shown to work effectively in extracting multiple signatures by applying multiple rounds of BC-IGLS. We refer to this algorithm as Multiple Signature IGLS (MS-IGLS). The multi signature extraction assumes an orthogonal, or nearly orthogonal, set of signatures, and progresses by finding one signature after another. If the multiple embeddings are done with equal embedding amplitudes, we find the first application of IGLS is equally likely to converge to any one of them. After that, the orthogonality property is used, and we use the Gram-Schmidt technique to orthogonalize each new signature iterate against all previously found signatures. Indeed, suppose that we have so far found signatures $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$. Then the MS-IGLS algorithm replaces Eq. (11) with

$$\mathbf{s}'_{k+1} = \frac{1}{N}\mathbf{Y}\mathbf{b}' - \sum_{i=1}^{k}\left(\frac{1}{N}\mathbf{b}'^{T}\mathbf{Y}^{T}\mathbf{s}_i\right)\mathbf{s}_i \tag{13}$$

which ensures the iterates $\mathbf{s}'_{k+1}$ are orthogonal to all previously found signatures $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$. Implementation details and test results are described below.

3. EXPERIMENTS AND RESULTS

The introduction to spread spectrum information hiding given above provides a firm foundation for information hiding in a variety of host data sets. The development also provides a clear prescription for fully blind extraction using the BC-IGLS algorithm, and for multi-signature extraction using the MS-IGLS algorithm, both presented for the first time here. The behavior of the blind extraction technique depends on the correlation structure in the host data, the number of hidden bits, the number of trial signatures used, and the initial signature guess used as IGLS seed.

3.1 Correlated Cover Data

The cover datasets we consider are created using a heuristic algorithm that adds a specified correlation structure to zero-mean Gaussian datasets. Although the DCT coefficients of JPG cover data provide a wider variety of behavior, these data sets provide an effective testing ground for our algorithms.

Our technique uses Cholesky decomposition, which allows us to specify (on average) the autocorrelation matrix of the cover data. Thus our study is not aimed at any particular cover data, but is instead intended to capture the behavior of the IGLS algorithm in this more general situation of weakly correlated host data.

Our algorithm for creating correlated host data starts by generating a real-valued positive-definite symmetric matrix $\mathbf{W}$ representing a weak correlation structure. This is accomplished using the following MATLAB commands:

```
W = abs(rand(L,L));

W = tril(abs(W - W') + eye(L));

W = W*W';
```

Once $\mathbf{W}$ is chosen, we can generate host data $\mathbf{X}$ with this desired autocorrelation matrix using the Cholesky decomposition $\mathbf{H}$ = chol($\mathbf{W}$), which by definition ensures $\mathbf{W} = \mathbf{H}^T\mathbf{H}$. Indeed, let $\tilde{\mathbf{X}}$ be an uncorrelated Gaussian host dataset, generated using the MATLAB function call `randn(L,N)`. Then the autocorrelation matrix of this host dataset is approximated by the identity matrix $\mathbf{R}_{\tilde{\mathbf{X}}} = \tilde{\mathbf{X}}\,\tilde{\mathbf{X}}^T = \mathbf{I}$. Now let $\mathbf{X}=\mathbf{H}^T\tilde{\mathbf{X}}$ be the transformed data. It is straightforward to show that the autocorrelation function for this new dataset is $\mathbf{W}$:

$$\mathbf{R}_{\mathbf{X}} = \mathbf{X}\,\mathbf{X}^T = \left(\mathbf{H}^T\tilde{\mathbf{X}}\right)(\mathbf{H}^T\tilde{\mathbf{X}})^T = \mathbf{H}^T\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{H} = \mathbf{H}^T\mathbf{R}_{\tilde{\mathbf{X}}}\mathbf{H} = \mathbf{H}^T\mathbf{I}\mathbf{H} = \mathbf{W}. \tag{14}$$

Note the identity $\mathbf{R}_{\tilde{\mathbf{X}}} = \mathbf{I}$ holds only in the limit of large N, so for finite N the relation $\mathbf{R}_{\mathbf{X}} = \mathbf{W}$ is only approximate.

3.2 General properties of IGLS Signature Extraction

In our first set of tests, we performed many individual runs of the IGLS algorithm using random initial signature guesses. Each random initial guess signature yields a "found" signature. With a known embedding signature, we determined whether IGLS was successful (i.e., converged to the correct signature) by comparing the BER obtained using the known signature to the BER using the signature IGLS found. In doing so, it is important to stay within the embedding amplitude regime such that the BER when using the correct signature differs significantly from 0.5. Thus we do not consider very low embedding amplitudes, where the BER is approximately 0.5 even when the correct signature is used. This region is the top of the familiar "waterfall plot" of BER vs. embedding amplitude. A successful IGLS run was thus determined as one yielding a signature that gives a BER sufficiently close to the BER obtained when using the known signature, yet sufficiently far from 0.5 so as to be useful. From these experiments we determined the fraction of runs that yielded the correct signature, henceforth called the convergence fraction. Experiments were performed for both binary and real-valued signatures. We only provide a brief qualitative summary of these results here.

For the case of binary signatures, our simulations show mostly expected behavior: convergence fractions are small at low embedding amplitudes, and increase toward unity as the embedding amplitude increases. The transition region

occurs at decreasing embedding amplitude as the signature length increases, again as expected. We also find the transition region occurs at lower embedding amplitude when less data is hidden, unless the number of hidden bits approaches the signature length, in which case the convergence fractions never deviate appreciably from zero.

For the case of real-valued embedding signatures, our results show an opposite dependence on signature length as with binary signatures, i.e., decreasing convergence fractions as signature length increases. However, we do not see convergence fractions increasing towards unity as the embedding amplitude increases. Instead there is a peak in convergence fraction, at embedding amplitudes that decrease as signature lengths increase. We also find, for equal embedding amplitudes, eigen-signatures yield significantly larger convergence fractions than random real-valued signatures.

3.3 Blind IGLS Extraction using BC-IGLS

The idea behind the BC-IGLS algorithm for blind signature extraction, as described above, is to run many IGLS trials using random signature guesses, and use distances between extracted signatures to identify when the algorithm has converged to the correct signature. The BC-IGLS claims successful blind recovery if a cluster of signatures is found whose pairwise distances are much smaller than the typical distance between extracted signatures.

Implementing the BC-IGLS algorithm requires choosing optimal values for both the minimum threshold distance for clustering, and the minimum cluster size. We find that a minimum cluster size of two is a practical choice balancing detection properties with computation time. This maximizes our chance of finding a cluster when convergence fractions are low, while maintaining a relatively low false positive rate (see below). Correct identification requires at least two trials to converge to the correct signature. For binary signatures, the optimal minimum distance threshold for clustering is found to be zero, corresponding to perfect signature matches. This is a natural criterion for binary signatures, where Hamming distance is used. For real-valued signatures, it seems that zero distance threshold would not be appropriate. However, we use zero distance, because the data iterates $\mathbf{b}'$ are still binary, and once the signature iterations from two trials get close enough to yield the same data set, the next iteration will yield exactly the same signature for both.

The BC-IGLS algorithm was tested by fixing N, L, and A, running multiple trials, and classifying the results using detection theory[3] metrics (1) true positive, (2) true negative, (3) false positive, and (4) false negative. These cases can be described as follows:

1. True Positive (TP): At least two trials converged to the correct signature, and the BC-IGLS algorithm correctly identified this cluster.
2. True Negative (TN): Less than two trials converged to the correct signature, and the BC-IGLS algorithm correctly reports no clusters.
3. False Positive (FP): Less than two trials converged to the correct signature, yet the BC-IGLS algorithm identified a cluster of signatures and reported this as the correct signature. This also includes the case where two or more trials did converge to the correct signature, but the algorithm found a larger cluster with the incorrect signature.
4. False Negative (FN): At least two trials converged to the signatures yielding BER close to the BER obtained using MMSE filtering, yet the BC-IGLS algorithm identified no clusters because they were not within the threshold distance.

Each of these cases is illustrated in Figure 1, which used binary signatures with parameter set $N = 500$, $L = 25$, and $A = 2$. The plots show the actual BER of the converged signature as a function of trial number. Trials in which the converged signature yielded BER that matches that expected from optimal MMSE filtering (shown as a red horizontal line, and very near zero in this case). If a cluster was found, the data points in the cluster are indicated by the red circles.

If no cluster was found, a red circle marks only trial #1. (It is important to recognize that two signatures may yield the same BER without being equal, which contributes to the existence of false negatives.

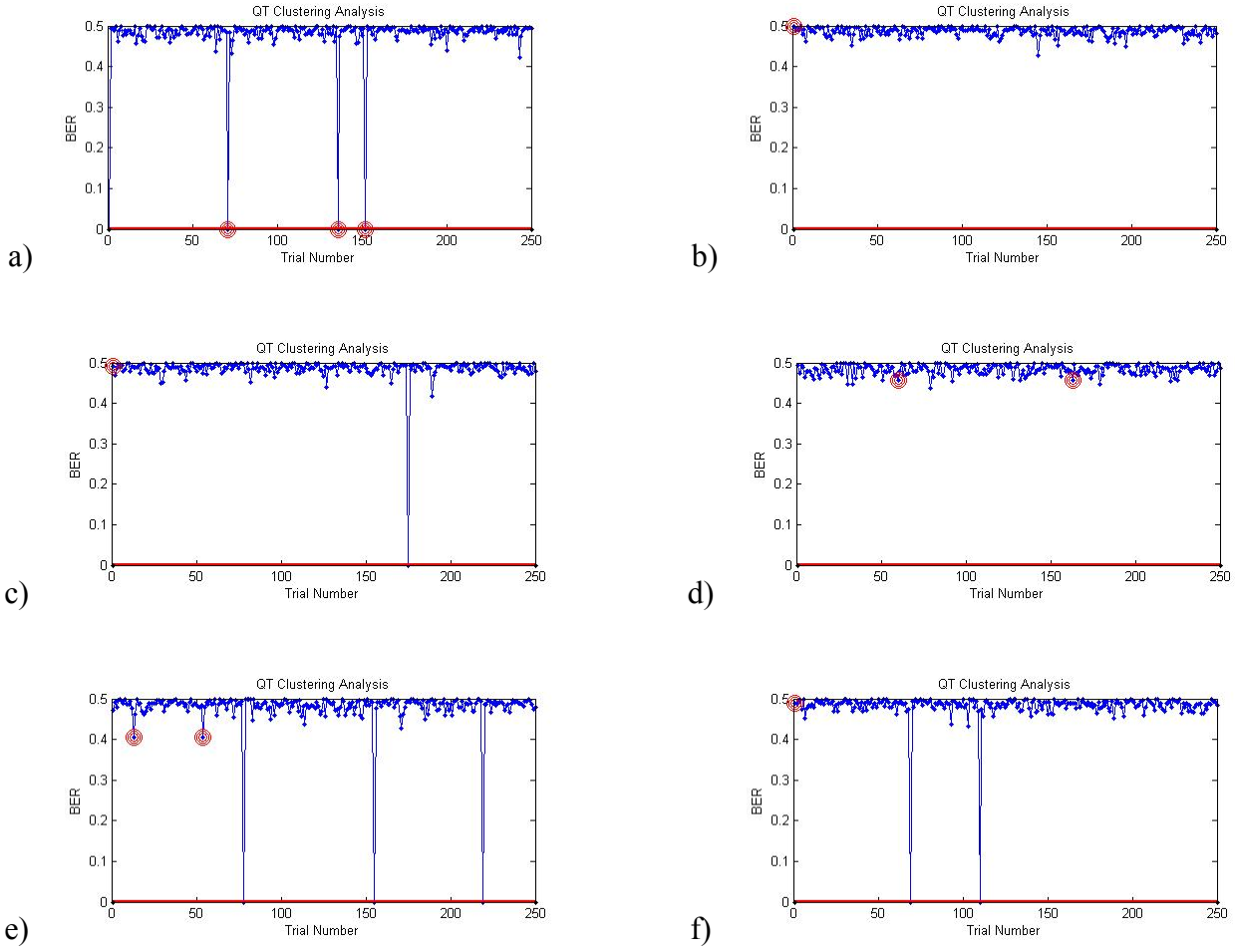

Figure 1: Examples of running the BC-IGLS blind extraction algorithm for 250 trials with $N = 500$, $L = 25$, $A = 2$, and using random binary signatures. Plot a shows a True Positive, plots b and c show true negatives, plots d and e show False Positives, and plot f shows a false negative.

We now present results that illustrate the behavior of the BC-IGLS algorithm using binary signatures, using these detection theory metrics. As described above, we use a minimum threshold of zero for clustering, and we use a minimum cluster size of two for identifying correct convergence.

Figure 2 shows results from a series of 6 runs. The detection metrics are plotted individually as a function of embedding amplitude. The 6 runs vary the number of trials T while holding the number of hidden data points $N = 200$ and the signature length $L = 25$ fixed.

We can see that the algorithm converges to the correct signature once the embedding amplitude exceeds 2 (3 dB). At the more interesting lower embedding amplitudes, the algorithm has a tendency to yield false positives for this relatively short signature, especially as the number of trials increases. This is because two or more matching signatures are likely, even when the algorithm did not yield the correct signature. When only 100 trials were made, true negatives dominated over false positives at low embedding amplitude, whereas with 600 trials, false positives dominated over true negatives.

This points out the disadvantage of simply increasing the number of trials in the run. In all cases some number of false negatives occurred at intermediate values of the embedding amplitude. As the number of trials increases, however, there is a clear trend toward fewer false negatives.

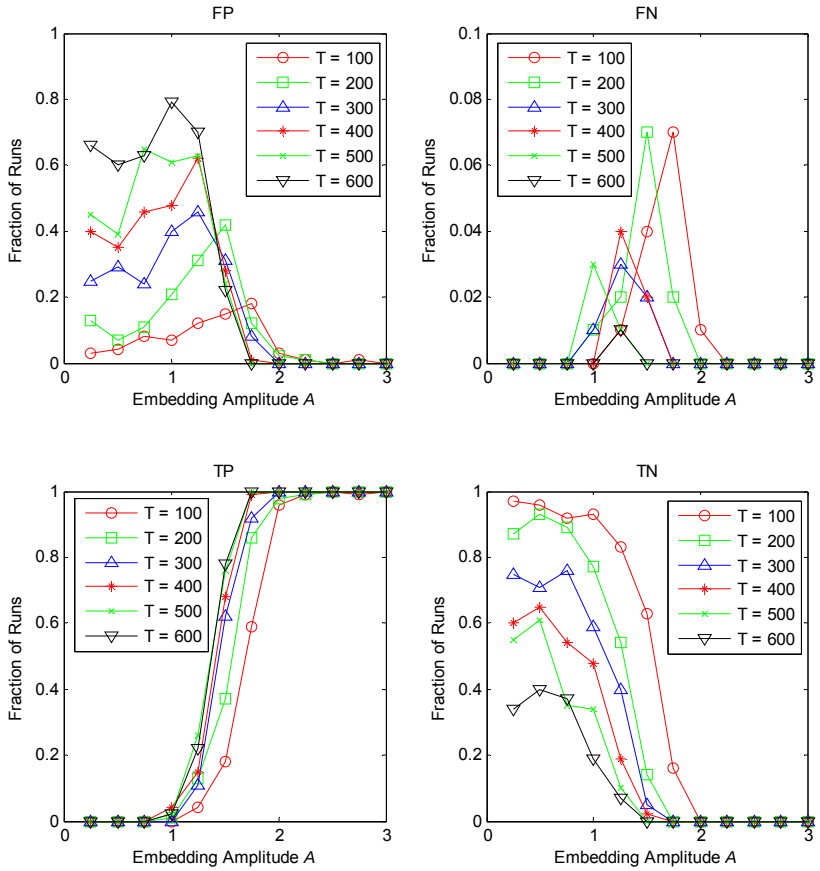

Figure 2: Detection theory metrics for a series of 6 runs of fully blind extraction using the BC-IGLS algorithm. The six runs had variable number of trials as shown, and fixed $N = 200$ and $L = 25$.

A similar series of runs was made at signature length $L = 10$, and essentially all trials yielded either false positive or true positive, the latter dominating at embedding amplitude above $A = 3$. This is a very short signature length, and the likelihood of false positives (where two or more signatures agree even without yielding to the proper hidden data) is even higher than for the case of signature length $L = 25$.

It is apparent the longer signatures should have less chance of yielding false positives at low embedding amplitude. This is tested explicitly in the series of runs shown in Figure 3, where 5 different signature lengths were run with fixed number of hidden bits $N = 1000$ and a relatively small number of IGLS trials $T = 100$. At the shortest signature length $L = 10$, false positives dominate. However excellent blind detection capabilities appear at all longer signature lengths, where true positives dominate at large embedding amplitudes, and true negatives dominate at low embedding amplitudes. As in the set of runs shown above, some false negatives appear in the transition region.

We can draw the following general conclusions regarding the performance of the BC-IGLS algorithm at fully blind detection, characterized in terms of the convergence fractions:

1. Successful blind extraction, characterized by a majority of true positives, is the dominant behavior at large embedding amplitudes, where convergence fractions are high.

2. Successful blind extraction extends to lower embedding amplitude as the number of trials T increases. However, at embedding amplitude below this threshold, increasing the number of trials also leads to a majority of false positives.
3. Longer signatures allow successful blind extraction at lower embedding amplitudes, avoiding false positives, as expected. However, the trade-off between TN and FP still exists as the number of trials increases.
4. Low convergence fractions are difficult to deal with, yielding a large fraction of false positive results where two or more signatures agree (i.e., false cluster), even though they do not match the actual embedding signature.

These results are significant as a practical guide in deploying blind detection techniques for spread spectrum hiding techniques. Although many variations to the basic algorithm could be developed and tested, the broad characteristics of the algorithm's behavior have been well-characterized as a function of embedding amplitude and signature length, the main variables under consideration.

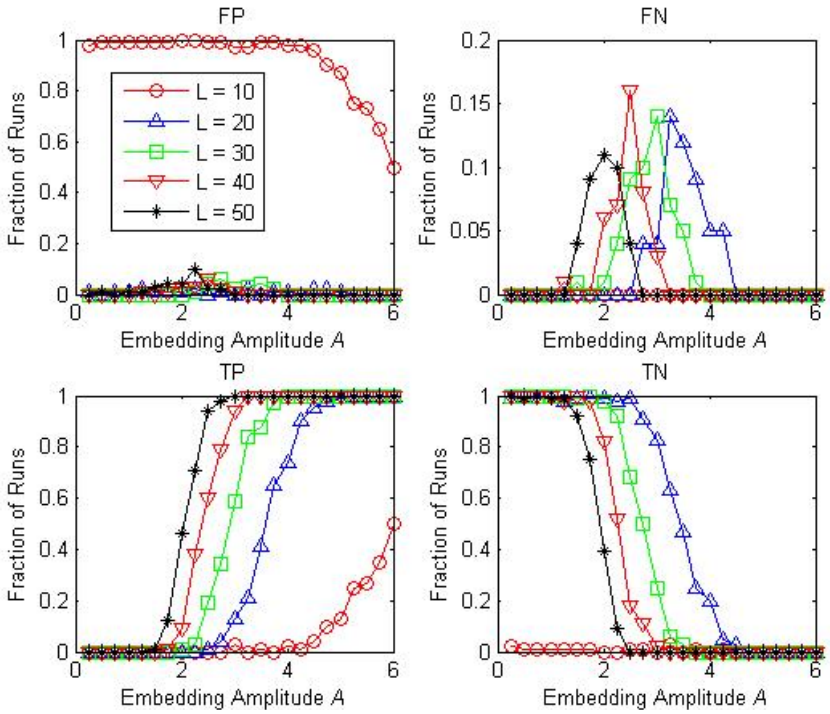

Figure 3: Detection theory metrics for a series of 6 runs of fully blind extraction using the BC-IGLS algorithm. The six runs had variable signature length as shown, and fixed $N = 200$ and number of trials $T = 100$.

Multi-Signature IGLS Algorithm (MS-IGLS)

The Multi-Signature IGLS algorithm (MS-IGLS) is intended to extract multiple data sets from a multi-user (or multi-signature) embedding. We present numerical results that demonstrate the operation of this algorithm for several different sets of parameter values.

The MS-IGLS algorithm performs BC-IGLS multiple times, each called a cycle. During each cycle, the BC-IGLS algorithm, modified to use Eq. (11), is used for obtaining the next signature iterate. Within each cycle, the modified BC-IGLS algorithm runs, returning a result indicating whether or not a signature was found, and, if so, what that signature was. Subsequent cycles will orthogonalize against all previously found signatures. For example, if during the first three cycles, the BC-IGLS algorithm returns a found signature only twice, the fourth cycle will orthogonalize against the two previously found signatures on the fourth cycle.

The success of the MS-IGLS algorithm requires correct identification of signatures at each cycle. False negatives essentially waste a cycle, causing the algorithm to need more cycles to extract the desired number of signatures. False positive results are more of a problem, however, because they cause subsequent runs to orthogonalize against the false positive, potentially limiting the ability to find proper signatures during subsequent cycles. Another consideration is that the MS-IGLS is designed for optimal performance when the signature set used for multiple signature hiding is an orthogonal set. This ensures that the orthogonalization step is not steering iterations away from a yet-to-be extracted signature during subsequent cycles.

Our tests of the MS-IGLS algorithm are set up as follows. We perform data hiding with 10 signatures, and run the MS-IGLS algorithm for 10 cycles in a fully blind situation. The algorithm will return up to 10 found signatures, one for each cycle. We then perform a data extraction using each of the found signatures, and compare these extracted data sets to all 10 of the known hidden data sets. Thus we calculate the BER of each found signature against all 10 known hidden data sets. Comparisons against the correct data set yields a low BER, whereas comparison to the other 9 data sets yields a BER of approximately 0.5. The result is a 10x10 matrix of BER values that identifies whether each cycle found an actual signature or not, and which one if it did.

Three examples are shown in Figures 4 to 6. Each figure shows two plots. The upper plot shows the 10x10 matrix of BER values described above, represented as grayscale values as shown in the legend. Columns represent MS-IGLS cycles (labeled "Runs" on the axes in the figures), and a dark square in a column indicates that one of the hidden data sets was successfully extracted. At the top of the grayscale legend is BER = 0.5, indicated by white. At the bottom we see BER = 0 indicated by black. The lower plot shows the extracted signatures (circles) and the actual signature (line). These signatures are offset vertically by the cycle number for clarity of presentation.

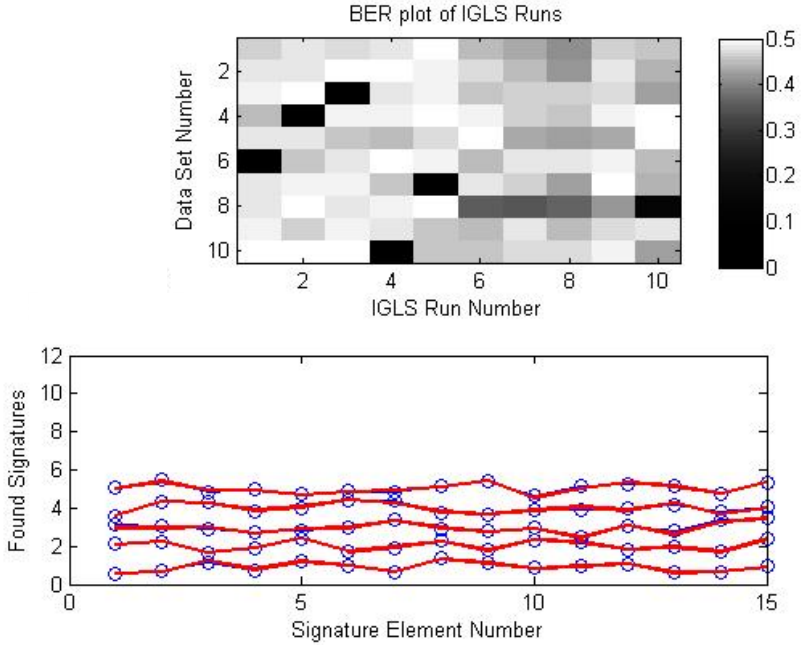

Figure 4: Multi signature extraction using MS-IGLS with 10 cycles and real-valued signatures.

The first, Figure 4, shows MS-IGLS applied to 10 real-valued signatures with constant embedding amplitude of 0 dB. By looking at the columns from left to right, we can see that cycle 1 converged to signature number 6, cycle 2 yielded signature 4, cycle 3 yielded signature 3, and so on. Out of 10 cycles, 5 yielded valid signatures. It appears to the eye that

cycle 10 yielded signature 8, but the BER was not low enough for the algorithm to mark it as "found". The lower plot shows the 5 signatures that were extracted, and there is clearly excellent agreement with the known signatures.

The next plot, Figure 5, shows MS-IGLS applied to eigen-signatures, again with constant embedding amplitude of 0 dB. This plot illustrates well the fact that eigen-signature extraction is much easier than random real-valued signatures, especially in regard to MS-IGLS, because all the signatures are orthogonal. We see that all 10 signatures were successfully extracted in the first 10 MS-IGLS cycles, and that the agreement with the known signatures is excellent.

We also repeated the experiment with eigen-signatures containing variable embedding amplitudes, as prescribed by the maximum capacity embedding condition. We find results similar to those of Figure 5, except the algorithm has a tendency to find the eigenvectors in order of embedding amplitude.

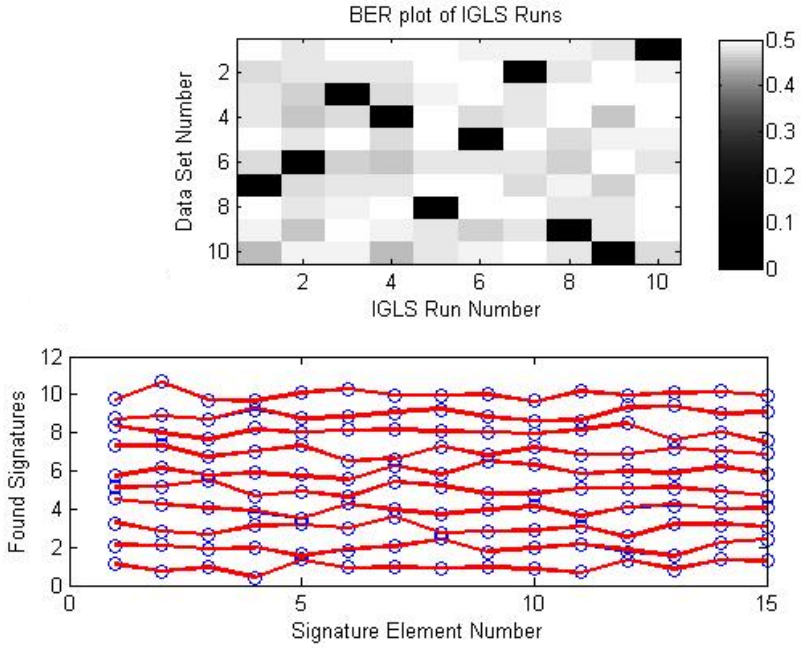

Figure 5: Multi signature extraction using MS-IGLS with 10 cycles.

We characterize the behavior of the MS-IGLS algorithm by looking at the average number of correct signatures extracted when run many times. All the runs use 10 signatures for 10 cycles. We repeat the experiment 100 times and report the average number of correct signatures found, out of 10. We repeat these experiments for several different parameter sets, as shown in the table below.

First, we see the performance for eigen-signatures is much better than for random real-valued signatures. This is attributable, at least in part, to the fact that random real-valued signatures do not in general form an orthogonal set, as do the eigen-signatures. MS-IGLS clearly works best when the signature set for multiple embedding is orthogonal. We see excellent results for eigen-signatures when the signature length is $L = 10$, with over 9 out of 10 signatures found, on average. As the signature size grows, we see fewer signatures successfully extracted, on average. This is attributable to the decreased convergence fractions, and points towards using a larger number of trials T to obtain better results with long signatures.

Comparing the eigen-signatures with uniform embedding to eigen-signatures with maximum capacity embedding amplitudes, we find MS-IGLS has on average better performance in the former case, except at the longest signature length studied. This may be expected because the uniform embedding amplitudes lend all signatures to detection,

whereas maximum capacity embedding has several signatures with low embedding amplitudes that are less likely to be extracted. The drop in performance at longer signatures is attributable to the effect of decreased convergence fractions as the signature length approaches the number of hidden bits (discussed previously). The drop in performance is more pronounced for the shorter $N = 200$ datasets than for the $N = 500$ data sets.

Table 1: Performance results for the MS-IGLS algorithm. All data shown were taken from runs using T = 500 trials and 10 embeddings.

Number of signatures correctly identified in first 10 MS-IGLS cycles, out of ten embeddings.						
	Eigen-signatures, uniform embedding amplitude		Eigen-signatures, maximum capacity embedding amplitude		Random real-valued signatures, uniform embedding	
	N = 200	N = 500	N = 200	N = 500	N = 200	N = 500
L = 10	9.69	10.00	9.08	9.84	4.98	6.63
L = 15	7.60	9.91	6.32	9.70	1.54	5.37
L = 20	0.07	6.02	3.56	8.34	0.02	2.25

These data give an overview as to where the algorithm works well, and where it has problems. All the runs were taken with $T = 500$ trials, and it is this value coupled with the corresponding convergence fraction that largely determines the success or failure of the MS-IGLS algorithm.

4. CONCLUSION

We have considered the use of spread spectrum techniques for information hiding into, and blind extraction from, correlated host data. This work constitutes a security analysis that quantifies the effectiveness of blind joint signature/data extraction from correlated host data. Our study is applicable to a variety of information hiding scenarios, including watermarking, authentication, annotations, and general covert communications. Our focus has been on performance evaluation of the recently introduced IGLS algorithm, and developing two extensions to the algorithm, that enable practical application.

The first extension is BC-IGLS, blind IGLS extraction using clustering. This algorithm is needed because the IGLS algorithm requires an initial signature guess, and will converge only for some fraction of signature guesses. The BC-IGLS algorithm seeks multiple trials, that yield the same, or nearly the same, signature result, and marks this as the correctly identified result. This algorithm assumes random initial signatures that do not converge to the proper signature yield random signatures as output. We have characterized the BC-IGLS algorithm and find that excellent behavior is attained with binary signatures with sufficient embedding amplitude. BC-IGLS performs much less satisfactorily when real-valued signatures are used, behavior characterized by low convergence fractions. This demonstrates the relative security of real-valued signatures when considering an IGLS-equipped adversary.

The second extension MS-IGLS performs multi signature extraction from multi user embedding. This new algorithm applies a modified BC-IGLS algorithm in multiple cycles, attempting to extract distinct embedding signatures each cycle by avoiding previously found signatures. This is accomplished by orthogonalizing against all previously found signatures at every IGLS iteration. Our tests show that the algorithm is effective when orthogonal sets of eigen-signatures are used and less effective when random real-valued signatures are used.

REFERENCES

[1] Cox, I. J., Kilian, J., Leighton, F. T. and Shannon, T., “Secure spread spectrum watermarking for multimedia,” IEEE Trans. Image Proc., 6(12):1673-1687, Dec. 1997.

[2] Proakis, J.G., “Digital Communications”, 4th Ed., McGraw Hill, 2001.

[3] Meyer, C.D., “Matrix Analysis and Applied Linear Algebra”, SIAM, 2000.

[4] Gkizeli, M., Pados, D., Batalama, S., and Medley, M., “Blind iterative Recovery of Spread-Spectrum Steganograpic Messages”, IEEE International Conference on Image Processing, 2005. vol. 2, pp. 1098-1101, Sept. 2005.

[5] Batalama, S. N., Medley, M. J., and Pados, D. A., ‘‘Robust adaptive recovery of spread- spectrum signals with short data records,’’ IEEE Transactions on Communications, vol. 48, pp. 1725-1731, Oct. 2000.

[6] Gkizeli, M., and Pados, D. A., "Image-adaptive spread-spectrum steganography,’’ Proc. 38th Conf. on Information Sciences and Systems (CISS 2004), Princeton, NJ, pp. 1570-1574, 2004.

[7] Li, M., Kulhandjian, M., Pados, D., Batalama, S., and Medley, M., “Passive Spread-Spectrum Steganalysis”, 2011 IEEE International Conference on Image Processing, paper TP.L5.8, 2011.

[8] Gkizeli, M., Pados, D., Medley, M., “Optimal Signature Design for Spread-Spectrum Steganography," IEEE Transactions on Image Processing, vol. 16, no.2, pp.391-405, Feb. 2007.

[9] Wei, L., Pados, D. A., Batalama, S. N., and Medley, M. J., ‘‘Sum-SINR/sum-capacity optimal multisignature spread-spectrum steganography,’’ in Proc. SPIE Mobile Multimedia/Image Processing, Security, and Applications Conf., SPIE Defense & Security Symposium, Orlando, FL, vol. 6982, pp. 0D_1-0D_10, Mar. 16-20, 2008.